統計量による$\alpha\beta$法の効率化 (計算モデルとアルゴリズム)

AUTHOR(S):

亀田, 純也; 笠井, 琢美

CITATION:

亀田, 純也 ...[et al]. 統計量による$\alpha\beta$法の効率化 (計算モデルとアルゴリズム). 数理解析研究所講究録 1999, 1093: 62-67

ISSUE DATE:

1999-04

URL:

http://hdl.handle.net/2433/62971

数理解析研究所講究録
1093巻 1999年 62-67

統計量による$\alpha\beta$法の効率化

亀田 純也 (Jyunya Kameda)　笠井 琢美 (Takumi Kasai)

電気通信大学大学院情報工学専攻
E-mail:kameda-j@calvyn.cs.uec.ac.jp

1 はじめに

ゲームにおける局面を節点に、手を枝に対応させたグラフをゲームの木という。ゲームにおけるすべての局面を調べることができれば良いのだが、余程単純なゲームでもない限り、局面の数が指数的に増加していくので探索は不可能である。そこで、ゲームの木を探索可能な深さで打ち切りその局面を評価関数により評価値 (有利な局面になればなるほど大きい値、不利な局面になればなるほど小さい値) を返す。この評価値を互いのプレイヤーが最善をつくすと仮定した探索を行い、一番大きい評価値を取ることのできるような手を決定する。

ゲームの木の探索において重要なのは、できる限り深くまで読むことであり、一般的に深く読めば読むほど局面の評価が正確になり強くなる。読む必要のない局面はできるだけ読まないようにして探索の効率を上げるのが基本である。一般に$\alpha\beta$法 [1] という効率のいい探索アルゴリズムが知られている。$\alpha\beta$法は、探索する必要の無い節点をα値、β値によって枝刈りを行い効率を上げるている。

本研究では統計量を用い評価値を予想し、その予想した値がα値、β値に比べ大幅に異なるとき、枝刈りされたものと考え枝刈りをして効率を上げるという方法を提案し、立体四目並べ[1]というゲームに対して実装を行った。$\alpha\beta$法で訪れる節点の数を統計量を用いカットを行ったときに訪れる節点の数で割ったものをスピードアップ率、この方法によって選ばれた手が$\alpha\beta$法で選ばれた手と同じ評価値をもつ割合を的中率として実験を行った。的中率99%でスピードアップ率3.0、また的中率81%でスピードアップ率40という結果となった。スピードアップ率は、実際の実行時間40倍を反映したものであり、的中率は、評価値自体がヒューリスティックなもので正確な局面の形勢を表わす値ではないので、実用上、十分な有効性を示すことができていると考えられる。

2 準備

ここでは、準備としてゲームとそれに対応するゲームの木、ゲームの木で探索して求める評価値になどについて定義する。

定義 2.1　ゲームとは、$\mathcal{G}=(Q,\Sigma,t,q_0)$のことである。ここで、$Q$は有限集合で、その元を**局面**と呼ぶ。$\Sigma$の各元$\sigma$に対し部分関数$f_\sigma : Q \to Q$が与えられている。以下では、$f_\sigma$を単に$\sigma$と書く。$\Sigma$の各元$\sigma$を**手**と呼ぶ。$t : Q \to \mathcal{R}$。$t$を**評価関数**と呼ぶ。$q_0$は$Q$の元で**初期局面**と呼ぶ。

定義 2.2　ゲームの木とは、$\mathcal{T}=(V,E,r,l)$のことである。ここで、(V,E,r)は、**根付き木**のことである。すなわち、(V,E)は閉路をもたない有限有向グラフで、Vの元を**節点**、Eの元を**枝**と呼び、rはVの元で**根**と呼ぶ。rに入る枝はなく、r以外の節点はただ1つの入る枝を持つ。$l : V \to Q$、$l : E \to \Sigma$。lをラベル付け関数と呼び、以下の条件を満たす。

1. $l(r)=q_0$

[1]Win95用の遊べる立体四目並べのプログラムは笠井研のホームページ http://calvyn.cs.uec.ac.jp に置いてあるのでダウンロードして試してみてほしい。

2. $v \in V$、$l(v) = q$とする。$\sigma(q)$ が定義されているならば、枝 $e = (v, u)$ が存在し $l(e) = \sigma$、$l(u) = \sigma(q)$

定義 2.3　$\mathcal{T} = (V, E, r, l)$ をゲームの木とする。v、u を $\mathcal{T}$の節点とする。このとき、vの深さとは、根 rから vまでの道の長さのことである。木の深さとは、根 rから葉までの深さの最大値のことである。

定義 2.4　$\mathcal{T} = (V, E, r, l)$ をゲームの木とする。qを局面、d を正の整数とする。qを初期局面とする深さ **d** のゲームの木 (以下これを (q, d) 木と呼ぶ) を次で定義する。
lが以下の条件を満たすもの。その他の条件はゲームの木の条件を満たす。

1. $l(r) = q$

2. $v \in V$、$l(v) = q$とする。vの深さが d 未満で$\sigma(q)$ が定義されてるならば、枝 $e = (v, u)$ が存在し $l(e) = \sigma$、$l(u) = \sigma(q)$。vの深さが d ならば、vは葉である。

定義 2.5　$\mathcal{T} = (V, E, r, l)$ をゲームの木、vを $\mathcal{T}$の節点とする。評価値 $T(v)$ を以下で定義する。

- vが葉のとき

$$T(v) \stackrel{\text{def}}{=} t(l(v))$$

- vが葉以外のとき

$$T(v) \stackrel{\text{def}}{=} \max_{\sigma \in \Sigma} \{ -t(\sigma(l(v))) \}$$

しかし、$T(v)$ の探索は実際的には不可能なので深さ d まで探索した評価値を考える。
定義 2.6　$\mathcal{T} = (V, E, r, l)$ を (q, d) 木、vを $\mathcal{T}$の節点とする。評価値 $T_d(v)$ を以下で定義する。

- vが葉のとき

$$T_d(v) \stackrel{\text{def}}{=} t(l(v))$$

- vが葉以外のとき

$$T_d(v) \stackrel{\text{def}}{=} \max_{\sigma \in \Sigma} \{ -t(\sigma(l(v))) \}$$

(q, d) 木は同型を除いて一意に定まることに注意して、局面 qの評価値を節点の評価値を使って定義する。
定義 2.7　qを局面、$\mathcal{T} = (V, E, r, l)$ を (q, d) 木とする。評価値 $T_d(q)$ を以下で定義する。

$$T_d(q) \stackrel{\text{def}}{=} T_d(r)$$

定義 2.8　$\mathcal{G} = (Q, \Sigma, t, q_0)$ をゲームとする。Q_mを以下で定義する。

- $Q_0 = \{q_0\}$

- $Q_{m+1} = \Sigma(Q_m) = \{\sigma(q) | q \in Q\}$

Q_mの元を m 手目の局面と呼ぶ。ゲームが階層的とは、$i \neq j$ならば $Q_i \cap Q_j = \phi$のことである。以後、ゲームは階層的であるとする。

3 立体四目並べ

縦、横に4本づつ等間隔に16本の杭を垂直に立てたものを盤とする。それぞれの杭には、石を4個まで突き刺して積み上げることができる。先手(黒)と後手(白)が交互に自分の石を置いていって、先に、縦・横・斜めのいずれかに一直線に石を並べたプレイヤーの勝ちというゲームである。ただし、16本の杭すべてに4個づつ石が置かれ置く場所がなくなり先手・後手とも石を一直線に並べられなかった場合は引き分けとする。勝ち負け、あるいは引き分けになった局面を終局面を呼ぶ。また石をおける場所を座標で表わす。

$[n]$ で $\{0、1、\cdots、n-1\}$を表わすとする。局面を次で定義する。

$$q : [4] \times [4] \times [4] \to \{黒、白、なし\}$$

初期局面は

$$q_0(i,j,k) = なし \quad (0 \leq i、j、k \leq 3)$$

となる。ここでi、j、kは、それぞれx座標、y座標、z座標を表わし、黒、白、なしは、それぞれ黒石、白石が置かれていること、および石が置かれていないことを表わす。

局面 p の石の数とは $|\{(i,j,k)|p(i,j,k)\neq なし、0 \leq i、j、k \leq 3\}|$ のことである。

Σを局面の集合とする。Σ上の関係Rを次で定義する。pRqのとき、pからqの正当な手が存在する。すなわち、杭$(i、j)$に石を刺す手(以下これを$\sigma_{i,j}$と呼ぶ)が許される$(0 \leq i、j \leq 3)$。

局面 p が終局面でなく、石の数が偶数のとき

- $p(i、j、0) = なし$ ならば $q(i、j、0) = 黒$
- $p(i、j、0)\neq なし$ かつ $p(i、j、1) = なし$ ならば $q(i、j、1) = 黒$
- $p(i、j、1)\neq なし$ かつ $p(i、j、2) = なし$ ならば $q(i、j、2) = 黒$
- $p(i、j、2)\neq なし$ かつ $p(i、j、3) = なし$ ならば $q(i、j、3) = 黒$

局面 p が終局面でなく、石の数が奇数のとき

- $p(i、j、0) = なし$ ならば $q(i、j、0) = 白$
- $p(i、j、0)\neq なし$ かつ $p(i、j、1) = なし$ ならば $q(i、j、1) = 白$
- $p(i、j、1)\neq なし$ かつ $p(i、j、2) = なし$ ならば $q(i、j、2) = 白$
- $p(i、j、2)\neq なし$ かつ $p(i、j、3) = なし$ ならば $q(i、j、3) = 白$

立体四目並べ $\mathcal{G} = (Q, \Sigma, t, q_0)$ を次のように定める。

Qは上記の局面とする。Σは $\{\sigma_{i,j}|0 \leq i、j \leq 3\}$ とする。q_0は上記の初期局面とする。ところで立体四目並べには、石を4個並べて一直線することのできる線分の数は76本ある。点数を一つの線分について自分の石が1、2、3、4個並んでいる場合それぞれ4、16、64、10000点、相手の石が1、2、3、4個並んでいる場合それぞれ-4、-16、-64、-10000点、それ以外の線分には0点を割り当てる。76本の線分の割り当てられた点数を合計したものを評価関数tとして用いる。ただし、評価値は自分が石を置くときに減ることがないため、奇数手目の局面、偶数手目の局面で上下してデータが扱いにくい。そこで、計算機どうしを対戦させ、実際に訪れる局面qからqの評価値$t(q)$のデータを取った。初期局面からm手目の局面の評価値の平均値をM_mとして$t(q)$の平均値を予想した。その予想値を用いて評価関数を$t(q) - M_m$のように補正することによってデータを扱い易くする。以後、この補正した評価関数を用いて実験を行う。

4 統計量による効率化

qをm手目の局面とする。$\alpha\beta$法は、局面qの評価値$t_d(q)$を探索するときに必要のない節点を下限(α値)、上限(β値)によってを枝刈りをすることによって探索の効率を上げたものである。$\alpha\beta$法で、ゲームの木を深さkの各節点v_kの評価値$t(v_k)$とその節点から葉まで探索したときの評価値$T_d(v_k)$の差のデータ(増加値を名づける)を取る。このデータの平均値を$T_{m,k}$として、$T_d(v_k)$を$T_{m,k}+t(v_k)$と予想する。その予想した評価値が、$\alpha\beta$法の上限と比較して高くなりすぎ(または、下限を比較して低くなりすぎ)枝刈りをされると予想されるとき、枝刈りされるものと考えて探索を打ち切り探索の効率を上げる。

5 実験

統計量を用いた方法でどのくらい$\alpha\beta$法の効率が上がるのかプログラムを作成し実験を行う。その指標として以下を定義する。

$$\text{スピードアップ率} \stackrel{\text{def}}{=} \frac{\alpha\beta\text{法で訪れる節点の数}}{\text{統計量による}\alpha\beta\text{法で訪れる節点の数}}$$

ただし、この方法を使うと正確な評価値を計算するとは限らない、そこで、

$$\text{的中率} \stackrel{\text{def}}{=} (\alpha\beta\text{法によって選ばれる手と同じ手を選ぶ割合})$$

を考える。

初期局面から4手ランダムに石を置いた局面を100個生成し、それぞれの局面から計算機どうしを対戦させデータを取った。統計量による方法は予想値(現在の局面の評価値$t(q)+T_{m,k}$)+カット値がα値より小さい、または、予想値−カット値がβ値より大きいときにカットを行う。以下にそれぞれのアルゴリズムを示す。

```
integer procedure αβ法(integer depth, 局面 p,integer alpha,integer beta):
  begin integer m, i, t, d;
    次の局面をすべて生成し q = σ(p)、
    −t(q) の値の大きい順番にソートし、それぞれ p_1, p_2, ···, p_d とする。
    if d = 0 または depth = 0 then αβ法 := t(p) else
    begin
      m := alpha;
      for i := 1 to d do
      begin
        t := −αβ法(depth − 1, p_i, −beta, −m);
        if t > m then m := t;
        if m ≥ beta then goto cut;
      end;
    cut: αβ法 := m;
    end;
  end;
```

図 1: $\alpha\beta$法

```
integer procedure 統計量によるαβ法(integer depth, 局面 p,integer alpha,integer beta):
  begin integer m,i,t,d;
    次の局面をすべて生成し q = σ(p)、
    −t(q) の値の大きい順番にソートし、それぞれ p1,p2,…,pdとする。
    if d = 0 または depth = 0 then 統計量によるαβ法 := t(p) else
    if (t(p) + Tm,k) + カット値 < alpha then 統計量によるαβ法 := alpha else
    if (t(p) + Tm,k) − カット値 > beta then 統計量によるαβ法 := beta else
    begin
      m := alpha;
      for i := 1 to d do
      begin
        t := −統計量によるαβ法(depth − 1, pi, −beta, −m);
        if t > m then m := t;
        if m ≥ beta then goto cut;
      end;
    cut: 統計量によるαβ法 := m;
    end;
  end;
```

図 2: 統計量によるαβ法

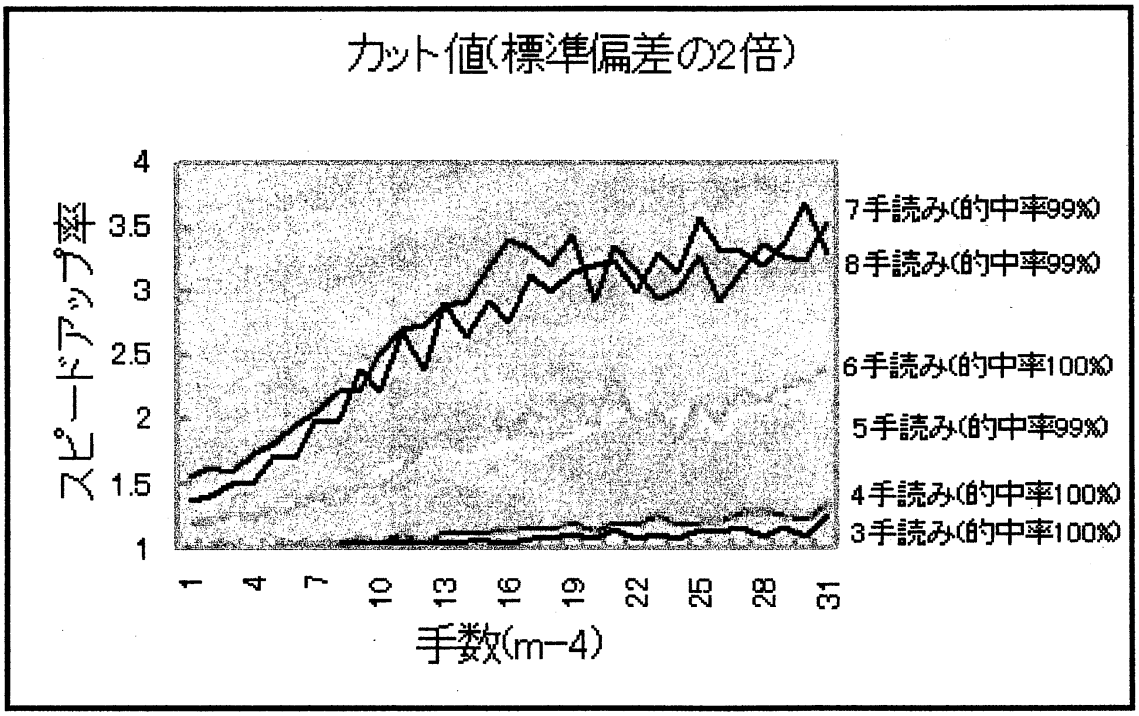

図 3: 結果 1

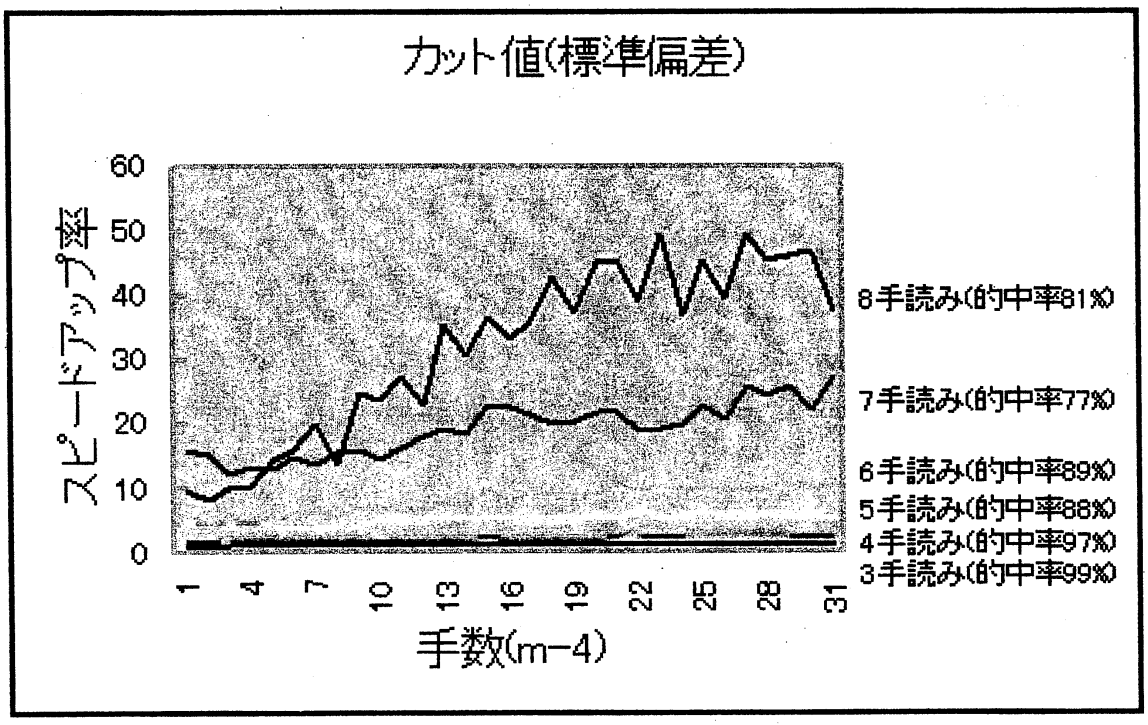

図 4: 結果 2

6 まとめ

カット値を標準偏差の 2 倍として実験を行うと 8 手読みで、的中率 99%、スピードアップ率 3.0 となり、カット値を標準偏差として実験を行うと 8 手読みで、スピードアップ率 40、的中率 81%となり、本稿で述べた方法の有効性を示すことができた。

実際のゲームにおいては、単純に葉にランダムな評価値が割り当てられたものではなく、親子、兄弟の節点の評価値には、強い相関がある。したがって、このようなゲームを形式化したモデルを使って本稿で述べた方法の理論的な解析をすべきであるが、形式化が困難であり、できていない。そのため、今のところ実験でしか有効性を確かめることができない。しかし、この統計量による方法は、立体四目並べというゲーム以外にも有効であると考えている。

参考文献

[1] Donald E.Knuth and Ronald W.Moore, *An Analaysis of Alpha-Beta Pruning*, Artificial Intelligence 6 (1975),293-326.

[2] 松原 仁・竹内郁雄 編, ゲームプログラミング, 共立出版,1997